## REMARKS

By this Amendment, claims 1-16 are amended. Thus, claims 1-16 are active in the application. Reexamination and reconsideration of the application are respectfully requested.

The specification and abstract have been carefully reviewed and revised to correct grammatical and idiomatic errors in order to aid the Examiner in further consideration of the application and to correct the informalities identified in item 1 on pages 2-3 of the Office Action. Further, the abstract was revised in order to delete numerical references to the drawings. The amendments to the specification and abstract are incorporated in the attached substitute specification and abstract. No new matter has been added.

Also attached hereto is a marked-up version of the substitute specification and abstract illustrating the changes made to the original specification and abstract.

The Applicants thank the Examiner for kindly that the application is in condition for allowance except for the informalities of the specification identified in item 1 on pages 2-3 of the Office Action. Minor editorial revisions were made to claims 1-16 in order to improve their U.S. form. The present amendment of claims 1-16 was not submitted earlier since the necessity of the amendments was not noticed earlier and since the Ex Parte Quayle Action was issued as the first Office Action on the merits. Therefore, the Applicants respectfully request entry of the present amendments to claims 1-16. Further, the Applicants submit that the revisions made to claims 1-16 were not to broaden or narrow the scope of protection for the present invention. Accordingly, the Applicants respectfully submit that claims 1-16, as amended, are in condition for allowance for the reasons identified in item 3 on pages 3-4 of the Office Action.

In view of the foregoing amendments and remarks, it is respectfully submitted that the present application is clearly in condition for allowance. An early notice thereof is respectfully solicited.

If, after reviewing this Amendment, the Examiner feels there are any issues remaining which must be resolved before the application can be passed to issue, it is respectfully requested that the Examiner contact the undersigned by telephone in order to resolve such issues.

Respectfully submitted,

Nobuhiko YASUI et al.

By: Jonathan R. Bowser

Jonathan R. Bowser
Registration No. 54,574
Attorney for Applicants

JRB/pth
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
May 28, 2004

TITLE OF THE INVENTION

# RENDERING DEVICE

## BACKGROUND OF THE INVENTION

### Field of the Invention

**[0001]** The present invention relates to rendering devices and, more specifically, to a rendering device which can be incorporated in a drive assistant device. In more detail, the rendering device generates a display image of an area around a vehicle based on an image that is captured by an image capture device fixedly placed in the vehicle.

### Description of the Background Art

**[0002]** The drive assistant device incorporating such a rendering device as described above has been actively researched and developed. A conventional-type drive assistant device is mounted in a vehicle, and generally includes an image capture device, a rudder angle sensor, a computing unit, a rendering device, and a display device. The image capture device is fixedly placed in a predetermined position in the vehicle, and ~~takes charge of~~ the image capture device is provided for capturing an image of an area that is defined by ~~its own~~ the viewing angle of the image capture device. The resulting image is ~~now~~ hereinafter referred to as a captured image. The rudder angle sensor is also fixed in a predetermined position in the vehicle, and detects to what degree

the steering wheel of the vehicle is turned. Based on the detection result, the computing unit calculates ~~a path~~ an estimated path for the vehicle to take. The rendering device then renders the estimated path on the captured image, and the image generated thereby is ~~such~~ a display image such as the one shown in FIG. 20. The display image is displayed on the display device.

**[0003]** With such a display image on the display device, a driver of the vehicle can know if his/her current steering will fit the vehicle in ~~the~~ a parking space without colliding into any obstacle in a close range of the driver's vehicle. If his/her steering is not appropriate, the estimated path is displayed out of the parking space in the display image. Therefore, the driver can appropriately adjust the rudder angle of the steering wheel.

**[0004]** There is another type of conventional drive assistant device exemplarily disclosed in Japanese Patent examined Publication No. 2-36417 (1990-36417). The drive assistant device additionally carries an active sensor for measuring a distance between the vehicle and an obstacle that is observed near the estimated path. Based on the measurement result provided by the active sensor, the computing unit determines which part of the estimated path is to be rendered on the captured image. The part ~~thus~~ which is determined to be rendered on the captured image is ~~now~~ hereinafter referred to as a rendering estimated path. In this manner, the rendering device accordingly renders on the captured image the rendering estimated path, which ends right

before the obstacle.

**[0005]** The ~~above~~ above-described conventional drive assistant devices ~~carry~~ have the following two problems ~~as follows~~. First, the estimated path is fixedly determined in color for display. Thus, even if the color is similar in tone to a predominant color of the display image, the color is unchangeable. Here, the predominant color is mainly determined by the road, for example, regardless of whether the road paved or not with asphalt. If this is the case, the driver finds it difficult to instantaneously locate the estimated path on the display image.

**[0006]** Second, the estimated path that is rendered in the display image is represented simply by lines, ~~failing~~ which fails to help the driver instantaneously perceive how far he/she can move the vehicle. More specifically, as shown in FIG. 21, a vehicle *Vusr* carrying the conventional drive assistant device is moving toward an obstacle *Vbst*. In this case, the vehicle *Vusr* first collides into a corner point *Pcnr* of the obstacle *Vbst*, not intersection points *Pcrg* of an estimated path *Pp* and the surface of the obstacle *Vbst*. ~~It~~ This means that the farthest point possible for the vehicle *Vusr* to move is the corner point *Pcnr* of the obstacle *Vbst*. As such, even if the estimated path is so rendered as to end immediately before the object, the second problem remains yet unsolved.

## SUMMARY OF THE INVENTION

**[0007]** Therefore, an object of the present invention is to provide a rendering ~~device, a~~ device which generates display image ~~being generated thereby~~ that shows an estimated path in an eye-catching manner for the driver to easily locate.

Another object of the present invention is to provide a rendering ~~device, a~~ device which generates display image generated ~~thereby being~~ that is indicative and helpful for the driver to know how far he/she can move the vehicle.

**[0008]** The present invention has the following features to attain the above-described objects ~~above~~.

**[0009]** A first aspect of the present invention is directed to a rendering device for generating a display image of an area around a vehicle for drive assistance. The rendering device comprises a reception part for receiving a current rudder angle of a steering wheel of the vehicle from a rudder angle sensor fixed ~~therein~~ in the vehicle; a derivation part for deriving an estimated path for the vehicle to take based on the rudder angle received by the reception part; and an image generation part for generating the display image based on a captured image which is captured by an image capture device fixed in the vehicle, and the estimated path that is derived by the derivation part. Here, in the display image, the estimated path is overlaid on an intermittent basis.

**[0010]** A second aspect of the present invention is directed to a rendering device for generating a display image of an area

around a vehicle for drive assistance. The rendering device comprises a first reception part for receiving a distance to an obstacle that is located around the vehicle from a measuring sensor placed in the vehicle; a first derivation part for deriving a farthest point for the vehicle to move based on the distance received by the first reception part; a second reception part for receiving a current rudder angle of a steering wheel of the vehicle from a rudder angle sensor fixed in the vehicle; a second derivation part for deriving an estimated path for the vehicle to take based on the rudder angle received by the second reception part; and an image generation part for generating the display image based on a captured image which is captured by an image capture device fixed in the vehicle, the farthest point derived by the first derivation part, and the estimated path derived by the second derivation part.

**[0011]** These and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** FIG. 1 is a block diagram showing the hardware structure of a rendering device *Urnd1* according to a first embodiment of the present invention;

FIG. 2 is a diagram showing a display image *Sout* generated

by a processor 1 of FIG. 1;

FIG. 3 is a diagram showing a position where an image capture device 4 of FIG. 1 is placed;

FIG. 4 is a diagram showing a captured image *Scpt* captured by the image capture device 4 of FIG. 1;

FIG. 5 is a flowchart showing the processing procedure of the processor 1 of FIG. 1;

**[0013]** FIG. 6 is a diagram showing a left-side trajectory *Pp1* and a right-side trajectory *Pp2* derived in step S6 in FIG. 5;

FIG. 7 is a diagram showing overlaying position data *Dsp* generated in step S7 in FIG. 5;

FIG. 8 is a diagram showing the display image *Sout* generated in step S8 in FIG. 5;

FIG. 9 is a diagram showing the display image *Sout* generated in step S15 in FIG. 5;

FIG. 10 is a block diagram showing the hardware structure of a rendering device *Urnd2* according to a second embodiment of the present invention;

**[0014]** FIG. 11 is a diagram showing a display image *Sout* generated by a processor 21 of FIG. 10;

FIG. 12 is a flowchart showing the processing procedure of the processor 21 of FIG. 10;

FIG. 13 is a block diagram showing the hardware structure of a rendering device *Urnd3* according to a third embodiment of the present invention;

FIG. 14 is a diagram showing a display image *Sout* generated by a processor 41 of FIG. 13;

FIGS. 15A and 15B are diagrams showing placement positions of active sensors 441 to 444 of FIG. 13;

**[0015]** FIG. 16 is a flowchart showing the processing procedure of the processor 41 of FIG. 13;

FIG. 17 is a diagram for demonstrating the process in step S43 in FIG. 16;

FIG. 18 is a diagram for demonstrating the process in step S44 in FIG. 16;

FIG. 19 is a detailed diagram showing an estimated region *Rpt* generated in step S410 in FIG. 16;

FIG. 20 is a diagram showing a display image displayed by a conventional drive assistant device; and

FIG. 21 is a diagram for explaining problems unsolvable by the conventional drive assistant device.

~~DESCRIPTION OF THE PREFERRED EMBODIMENTS~~ DETAILED DESCRIPTION OF THE INVENTION

**[0016]** FIG. 1 is a block diagram showing the hardware structure of a rendering device *Urnd1* according to a first embodiment of the present invention. In FIG. 1, the rendering device *Urnd1* includes a processor 1, a program memory 2, and a working area 3. The program memory 2 is typified by ROM (Read Only Memory), and stores a program *PGa* for defining the processing procedure

in the processor 1. By following the program *PGa*, the processor 1 generates ~~such~~ a display image such as the display image *Sout* as shown in FIG. 2. The display image *Sout* shows ~~a~~ an estimated path *Pp* ~~estimated~~ for a vehicle *Vusr* (see FIG. 3) to take in the course of time. The estimated path *Pp* is composed of a left-side trajectory *Pp1* and a right-side trajectory *Pp2* ~~indicated by, respectively,~~ which are indicated by indicators *Sind1* and *Sind2*, respectively. Here, the left-side trajectory *Pp1* is for a left-rear wheel of the vehicle *Vusr*, while the right-side trajectory *Pp2* ~~by~~ is for a right-rear wheel of the vehicle. Further, the indicators *Sind1* and *Sind2* are both objects in a predetermined shape (e.g., circle, rectangle) that is previously stored in the program memory 2.

**[0017]** The working area 3 is typified by RAM (Random Access Memory), and used when the processor 1 executes the program *PGa*. The rendering device *Urnd1* ~~in~~ according to the ~~above~~ above-described structure is typically incorporated in a drive assistant device *Uast1*. The drive assistant device *Uast1* is mounted in the vehicle *Vusr*, and includes at least one image capture device 4, a rudder angle sensor 5, and a display device 6 together with the rendering device *Urnd1*.

**[0018]** As shown in FIG. 3, the image capture device 4 is embedded in the rear-end of the vehicle *Vusr*, and captures an image covering an area to the rear of the vehicle *Vusr*. The resulting image is a captured image *Scpt* as shown in FIG. 4. The rudder angle sensor

5 detects a rudder angle $\theta$ of the steering wheel of the vehicle ~~Vusr~~ *Vusr*, and transmits ~~it~~ the rudder angle $\theta$ to the processor 1. The rudder angle $\theta$ here indicates at what angle the steering wheel is turned with respect to the initial position. The steering wheel is considered to be in the initial position when the steering wheel is not turned, that is, when the vehicle *Vusr* is in the straight-ahead position. The display device 6 is typically a liquid crystal display.

**[0019]** Described next is the operation of ~~such~~ the drive assistant device *Uast1*. When the driver wants assistance ~~by~~ from the drive assistant device *Uast1*, the processor 1 starts executing the program *PGa*.

**[0020]** Refer now to a flowchart ~~of~~ in FIG. 5 for the processing procedure in the processor 1 written in the program *PGa*. In FIG. 5, the processor 1 first generates an image capture instruction *Icpt*, and transmits ~~it~~ the image capture instruction *Icpt* to the image capture device 4 (step S1). Here, as shown in FIG. 5, the procedure returns to step S1 after step S10 is ~~through~~ completed, and the processor 1 generates another image capture instruction *Icpt*. The program *PGa* is ~~so~~ written so that a time interval between those two image capture instructions *Icpt* is substantially a *t1* second. Here, the value of *t1* is ~~so~~ selected so as to allow the display device 6 to display the display image *Sout* for 30 frames per second. Herein, the image capture instruction *Icpt* is a signal instructing the image capture device 4 for image capturing. The

image capture device 4 responsively captures ~~such~~ a captured image *Scpt* such as shown in FIG. 4, and stores ~~it~~ the captured image *Scpt* in frame memory (not shown) reserved in the working area 3 (step S2).

**[0021]** The processor 1 then watches a deriving timing *T1* (step S3). This deriving timing *T1* is previously written in the program *PGa*, and allows the processor 1 to derive the left- and right-side trajectories *Pp1* and *Pp2* once every *t2* second. The value of *t2* is selected to be larger than that of *t1* (e.g., 0.1 second) since a change on a time base in the rudder angle $\theta$ is small.

**[0022]** In the deriving timing *T1*, the processor 1 generates a detection instruction *Idtc*, and transmits the detection instruction *Idtc* ~~it~~ to the rudder angle sensor 5 (step S4). The detection instruction *Idtc* is a signal instructing the rudder angle sensor 5 to detect the rudder angle $\theta$. The rudder angle sensor 5 responsively detects the rudder angle $\theta$, and stores ~~it~~ the rudder angle $\theta$ in the working area 3 (step S5).

**[0023]** Based on ~~thus~~ the detected rudder angle $\theta$, the processor 1 derives the left- and right-side trajectories *Pp1* and *Pp2* (step S6). More specifically, derived by the processor 1 here are equations respectively for the left- and right-side trajectories *Pp1* and *Pp2* under ~~the~~ Ackermann's model. Here, in the strict sense, the left- and right-side trajectories *Pp1* and *Pp2* are defined as being trajectories that are traced by left- and right-rear wheels of the vehicle *Vusr* on the condition that

the driver keeps the steering wheel at the currently derived rudder angle $\theta$. The left-side trajectory *Pp1* that is calculated by such an equation becomes an arc in a predetermined length. In more detail, the arc is a segment of a circle which is traceable by the vehicle ~~Vusr~~ *Vusr* around a ~~circling~~ center of the circle. The radius of the circle is equal to a distance from the ~~circling~~ center of the circle to a point having a rotation center of the left-rear wheel projected onto the road surface. The equation for the right-side trajectory *Pp2* is similar except that the arc is traced by the right-rear wheel, on its rotation center, of the vehicle *Vusr*.

**[0024]** Then, the processor 1 generates overlaying position data *Dsp* indicating where to overlay the two indicators *Sind1* and *Sind2*, and stores the data *Dsp* in the working area 3 (step S7). As an example, ~~if derived in step S6 are such~~ the left- and right-side trajectories *Pp1* and *Pp2* as shown in FIG. 6 are derived in step S6, the processor 1 calculates two points *a0* and *b0* ~~being~~ which are closest to the vehicle *Vusr* (not shown) on those trajectories *Pp1* and *Pp2*, respectively. The processor 1 then calculates a point *a1* ~~being away by~~ which is a predetermined distance $\Delta$d away from the point *a0* on the left-side trajectory *Pp1*, and a point *b1* ~~being away~~ which is also ~~by~~ the predetermined distance $\Delta$d away from the point *b0* on the right-side trajectory *Pp2*. The processor 1 repeats the same processing until *i* (where *i* is a natural number being 2 or larger) sets of coordinates such as (*a0*, *b0*), (*a1*, *b1*), ...,

(*a*(*i*-*1*), *b*(*i*-*1*)) are calculated. The sets of coordinates are numbered starting from the one closest to the vehicle *Vusr*. Accordingly, as shown in FIG. 7, ~~stored in the working area 3 is~~ the overlaying position data *Dsp* including those numbered sets of coordinates are stored in the working area 3.

**[0025]** Based on the overlaying position data *Dsp* and the aforementioned captured image *Scpt*, the processor 1 then generates a frame of the display image *Sout* on the frame memory (step S8). Here, as already described ~~by referring~~ with reference to FIG. 2, the display image *Sout* is the one having the indicators *Sind1* and *Sind2* overlaid on the captured image *Scpt*. In step S8, ~~more in~~ in more detail, the processor 1 first selects, from the overlaying position data *Dsp* generated in step S7, a set of coordinates which is not yet selected and which are the smallest in number. In this example, since ~~no~~ a set has not yet been selected, ~~selected now is~~ the set of (*a0*, *b0*) is now selected. The processor 1 then overlays the indicators *Sind1* and *Sind2* onto the points *a0* and *b0* in the captured image *Scpt* on the frame memory. After this overlaying process, such a display image *Sout* as the one shown in FIG. 8 is generated for one frame on the frame memory.

**[0026]** The processor 1 then transfers the display image *Sout* on the frame memory to the display device 6 ~~for display~~ to be displayed thereon (step S9). In the current display image *Sout* on the display device 6, the indicator *Sind1* is overlaid on the point *a0* on the left-side trajectory *Pp1*, and the indicator *Sind2*

is overlaid on the point *b0* on the right-side trajectory *Pp2*.

**[0027]** Then, the processor 1 determines whether ~~now~~ it is now the time to end the processing of FIG. 5 (step S10). If ~~determined not yet~~ the processor 1 determines that the processing should not end, the procedure returns to step S1 for generating another display image *Sout*. By the time ~~when~~ steps S1 and S2 are ~~through~~ completed, another captured image *Scpt* is newly stored on the frame memory. Then, in step S3, if the processor determines ~~determining~~ that the timing *T1* has not come yet, the processor 1 then watches a timing *T2* to change the overlaying positions of the indicators *Sind1* and *Sind2* (step S11). Here, the changing timing *T2* is previously written in the program *PGa*, and allows the processor 1 to change the overlaying positions of the indicators *Sind1* and *Sind2* once every *t3* second. If the value of *T3* is set too small, the indicator *Sind1* moves too fast from the point *a0* to *a1* for the driver to follow with her/his eyes on the display device 6. With consideration therefor, the value of *t3* is selected to be larger than that of *t1* (e.g., 0.05 second).

**[0028]** If the processor 1 determines that the timing *T2* has not come yet, the processor 1 generates a frame of the display image *Sout* on the frame memory (step S12). This is based on the captured image *Scpt* stored in step S2 and the set of coordinates currently selected in the overlaying position data *Dsp* (in this example, the set of (*a0*, *b0*)). As such, the resulting display image *Sout* is also the one having the indicators *Sind1* and *Sind2*

overlaid on the points *a0* and *b0* on the captured image *Scpt*. Then, the processor 1 transfers ~~thus~~ the generated display image *Sout* on the frame memory to the display device 6 ~~for display~~ to be displayed thereon (step S13).

**[0029]** Next, in step S10, if the processor 1 determines that ~~now~~ it is now not the time to end the processing of FIG. 5, the procedure returns to step S1. By the time when steps S1 and S2 are ~~through~~ completed, another captured image *Scpt* is newly stored on the frame memory. Then, in step S3, if the processor 1 determines that the timing *T1* has not come yet, and in step S11, if the processor 1 determines that the timing *T2* is now right, the procedure goes to step S14. Then, the processor 1 selects, from the overlaying position data *Dsp* on the working area 3, a set of coordinates which is not yet selected and which are the smallest in number (step S14). Since the set which was last selected ~~last~~ is (*a0*, *b0*), ~~selected this time is~~ the set (*a1*, *b1*) is now selected.

**[0030]** Next, the processor 1 generates a new frame of the display image *Sout* on the frame memory based on the captured image Scpt and the set of ~~coordinate~~ coordinates (in this example, the set of (*a1*, *b1*)) currently selected in the overlaying position data *Dsp* (step S15). As such, as shown in FIG. 9, the resulting display image *Sout* is the one having the indicators *Sind1* and *Sind2* overlaid on the points *a1* and *b1* on the captured image *Scpt*. Then, the processor 1 transfers ~~thus~~ the generated display image *Sout* on the frame memory to the display device 6 ~~for display~~ to be displayed

thereon (step S16).

**[0031]** Such steps S1 to S16 are repeated until the determination in step S10 becomes Yes to end the processing of FIG. 5. In this manner, the overlaying positions of the indicators *Sind1* and *Sind2* change, in increments of the predetermined distance $\Delta d$, from the points *a0* and *b0* to *a(i-1)* and *b(i-1)*, respectively. Thus, the indicators *Sind1* and *Sind2* are displayed as if moving in the same ~~heading~~ direction as the vehicle *Vusr* is heading towards along the left- and right-side trajectories *Pp1* and *Pp2*. ~~What is good~~ Advantageously, as the those indicators *Sind1* and *Sind2* are displayed on an intermittent basis, the left- and right-side trajectories *Pp1* and *Pp2* are also displayed on an intermittent basis on the display device 6. Accordingly, the left- and right-side trajectories *Pp1* and *Pp2* become more noticeable and are emphasized to a further degree. With such indicators *Sind1* and *Sind2*, the driver can instantaneously locate the trajectories *Pp1* and *Pp2* in the display image *Sout*.

**[0032]** Further, every time the rudder angle $\theta$ ~~comes from~~ is detected by the rudder angle sensor 5 according to the deriving timing *T1*, the processor 1 derives the left- and right-trajectories *Pp1* and *Pp2* based on the current rudder angle $\theta$. In this manner, the trajectories *Pp1* and *Pp2* displayed on the display device 6 always become ~~always~~ responsive to the driver's steering.

**[0033]** Note that, in the first embodiment, the changing timing *T2* may be variable. For example, in the case ~~that~~ where the

overlaying positions of the indicators *Sind1* and *Sind2* are relatively close to the vehicle *Vusr*, the program *PGa* may be ~~so~~ written so that the changing timing *T2* comes earlier. If so, the left- and right-side trajectories *Pp1* and *Pp2* become easier to notice.

**[0034]** Further, in the first embodiment, the predetermined distance $\Delta d$ between two successive points of $aj$ and $a(j+1)$ is constant on the left-side trajectory *Pp1*. Here, the value $j$ is a positive integer between 0 and $(i-1)$. The predetermined distance $\Delta d$ may not necessarily be constant. For example, in the case ~~that~~ where the point $aj$ is relatively close to the vehicle *Vusr*, the program *PGa* may be ~~so~~ written so that the predetermined distance $\Delta d$ is set to be relatively small so as to cause the processor 1 to select the point $a(j+1)$. Conversely, the program *PGa* may be ~~so~~ written so that the predetermined distance $\Delta d$ is set to be relatively large so as to cause the processor 1 to select the point $a(j+1)$. In both cases, the left- and right-side trajectories *Pp1* and *Pp2* become conspicuous to a further degree.

**[0035]** FIG. 10 is a block diagram showing the hardware structure of a rendering device *Urnd2* according to a second embodiment of the present invention. In FIG. 10, the rendering device *Urnd2* includes a processor 21, a program memory 22, and a working area 23. The program memory 22 is typified by ROM (Read Only Memory), and stores a program *PGb* for defining the processing procedure in the processor 21. By following the program *PGb*, the processor

21 generates ~~such~~ a display image *Sout* such as the one shown in FIG. 11. The display image *Sout* shows an estimated path *Pp* of the vehicle *Vusr* (see FIG. 3) to be traced by a left-rear wheel of the vehicle *Vusr*. The estimated path *Pp* is displayed only during a display time period *Pdt*, which will be ~~later described~~ described later.

**[0036]** The working area 3 is typified by RAM (Random Access Memory), and is used when the processor 21 executes the program *PGb*. The rendering device *Urnd2* ~~in~~ according to the ~~above~~ above-described structure is typically incorporated in a drive assistant device *Uast2*. Here, as to the drive assistant device *Uast2*, the only structural difference from the drive assistant device *Uast1* of the first embodiment is ~~including~~ that the drive assistance *Uast2* includes the rendering device *Urnd2* instead of the rendering device *Urnd1*. Thus, any component ~~appeared~~ illustrated in FIG. 1 ~~is under~~ has the same reference numeral in FIG. 10, and therefore is not described again.

**[0037]** Described next is the operation of ~~such~~ the drive assistant device *Uast2*. When the driver wants assistance ~~by~~ from the drive assistant device *Uast2*, the processor 21 starts executing the program *PGb* in the program memory 22.

**[0038]** Refer now to a flowchart ~~of~~ in FIG. 12 for the processing procedure in the processor 21 written in the program *PGb*. Compared with FIG. 5, the flowchart of FIG. 12 includes the same steps, and thus those ~~are under~~ steps having the same step numbers are

identical and thus are not described again.

**[0039]** First, by going through steps S1 to S6, the processor 21 derives an equation for the estimated path *Pp*. The procedure then goes to step S21, and the processor 21 generates the display image *Sout* based on the captured image *Scpt* stored in step S2 and the estimated path *Pp* derived in step S6. More specifically, the processor 21 renders ~~thus~~ the derived estimated path *Pp* in its entirety on the display image *Sout*, and the resulting display image *Sout* looks ~~as~~ like the one shown in FIG. 11.

**[0040]** The procedure then goes to step S9, and the processor 21 transfers the display image *Sout* currently on the frame memory to the display device 6 ~~for display~~ to be displayed thereon. Then, the processor 21 determines whether ~~now~~ it is now the time to end the processing of FIG. 12 (step S10), and if ~~not yet~~ the processor 1 determines that the processing should not end now, the procedure returns to step S1 for generating another display image *Sout* on the frame memory. By the time when steps S1 and S2 are ~~through~~ completed, another captured image *Scpt* is newly stored on the frame memory. Then, in step S3, if the processor 1 determines that the timing *T1* has not come yet, the processor 1 then determines whether it is now ~~is~~ in the display time period *Pdt* for the estimated path *Pp* (step S22). Here, the display time period *Pdf* is previously written in the program *PGb*, and comes every *t4* second in ~~this~~ the second embodiment. ~~It~~ This means that the estimated path *Pp* appears on and disappears from the display with a time lapse of

*t4* second. Note that, if the value of *t4* is set too small, the appearance and disappearance of the estimated path *Pp* will be too swift for the driver to notice. With consideration therefor, the value of *t4* is selected to be larger than that of *t1* (e.g., 0.1 second).

**[0041]** If the processor 21 determines that it is now ~~is~~ in the display time period *Pdt*, the procedure goes to step S21. The processor 21 then generates, on the frame memory, the display image *Sout* including the estimated path *Pp* (see FIG. 11). The procedure then goes to step S9, and the processor 21 transfers the current display image *Sout* on the frame memory to the display device 6 ~~for display~~ to be displayed thereon. Then, the processor 21 determines whether it is now ~~is~~ the time to end the processing of FIG. 12 (step S10), and if ~~not yet~~ the processor 1 determines that the processing should not end now, the procedure returns to step S1 for generating another display image *Sout*. In step S3, if the processor 21 determines that the deriving timing *T1* has not come yet, and in step S22, if the processor 1 determines that ~~now~~ the present time is not in the display time period *Pdt*, the procedure goes to step S23. In step 23, the processor 21 transfers, to the display device 6 ~~for display~~ to be displayed thereon, the captured image *Scpt* stored in step S2 (see FIG. 4) as the display image *Sout* without any ~~change~~ changes thereto (step S23).

**[0042]** Such steps S1 to S23 are repeated until the determination in step S10 becomes Yes to end the processing of FIG. 12. In this

manner, the estimated path *Pp* is displayed only during the display time period *Pdt*. The estimated path *Pp* appears on and disappears from the display on an intermittent basis. Accordingly, the estimated path *Pp* becomes noticeable, and the driver finds it easy to locate the estimated path *Pp* in the display image *Sout*.

**[0043]** FIG. 13 is a block diagram showing the hardware structure of a rendering device *Urnd3* according to a third embodiment of the present invention. In FIG. 13, the rendering device *Urnd3* includes a processor 41, a program memory 42, and a working area 43. The program memory 42 is typified by ROM (Read Only Memory), and stores a program *PGc* for defining the processing procedure in the processor 41. By following the program *PGc*, the processor 41 generates ~~such~~ a display image *Sout* such as the one shown in FIG. 14. The display image *Sout* shows an estimated region *Rpt* on a road surface *Frd* for the vehicle *Vusr* (see FIG. 3) to move. Specifically, the estimated region *Rpt* is defined by the left- and right-side trajectories *Pp1* and *Pp2* described above in the first embodiment, and a line segment *Llmt* passing through a no-go point *Plmt*. Here, the no-go point *Plmt* is a point indicating the farthest limit for the vehicle *Vusr* to move, and if the vehicle *Vusr* keeps moving, ~~it~~ the vehicle might first collide into the obstacle *Vbst* ~~first~~.

**[0044]** The working area 43 is typified by RAM (Random Access Memory), and is used when the processor 41 executes the program *PGc*. The rendering device *Urnd3* ~~in~~ according to the ~~above~~

above-described structure is typically incorporated in a drive assistant device *Uast3*. Here, as to the drive assistant device *Uast3*, the structural difference ~~from~~ between the drive assistant device *Uast1* and the drive assistant device *Uast3* is that the drive assistant device *Uast3* includes ~~is including~~ the rendering device *Urnd3* instead of the rendering device *Urnd1*, and further ~~including 4~~ includes four active sensors 441 to 444, which is exemplified ~~for~~ herein as a measuring sensor ~~in Claims~~. These are the only structural differences, and thus any component ~~appeared~~ illustrated in FIG. 1 ~~is under~~ has the same reference numeral in FIG. 13, and therefore is not described again.

**[0045]** As shown in FIG. 15A, the active sensors 441 to 444 are embedded in the rear-end of the vehicle *Vusr*, preferably, in a lateral direction. The active sensors 441 to 444 thus arranged ~~as such~~ emit ultrasonic waves or radio waves toward the area to the rear of the vehicle *Vusr*, and monitor reflected waves. Thereby, as shown in FIG. 15B, distances *d1* to *d4* to an obstacle *Vbst* located closest behind the vehicle *Vusr* are detected by the active sensors 441 to 444.

**[0046]** Described next is the operation of ~~such~~ the drive assistant device *Uast3*. When the driver wants assistance ~~by~~ from the drive assistant device *Uast3*, the processor 41 starts executing the program *PGc* in the program memory 42.

**[0047]** Refer now to a flowchart ~~of~~ in FIG. 16 for the processing procedure in the processor 41 written in the program *PGc*. In FIG.

16, the processor 41 first generates a distance measuring instruction *Imsr*, and transmits ~~it~~ the distance measuring instruction *Imsr* to all of the active sensors 441 to 444 (step S41). Here, the distance measuring instruction *Imsr* is a signal to instruct all of the active sensors 441 to 444 to detect the distances *d1* to *d4*, and to transmit those distances to the processor 41. The active sensors 441 to 444 each responsively perform such detection, and store the resultant distances *d1* to *d4* to the working area 43 (step S42).

**[0048]** Next, based on the ~~thus~~ detected distances *d1* to *d4*, the processor 41 calculates coordinates (*x1*, *y1*) to (*x4*, *y4*) of four points *P1* to *P4* on the surface of the object *Vbst* (step S43). Referring to FIG. 17, the process in step S43 is described in detail. FIG. 17 shows the vehicle *Vusr*, the obstacle *Vbst*, and a two-dimensional (2D) coordinate system. In the 2D coordinate system, the Y-axis connects a rotation center of a left-rear wheel *Wr1* and that of a right-rear wheel *Wr2*. With respect to the Y-axis, the X-axis is a perpendicular bisector parallel to a horizontal plane. As described above, the active sensors 441 to 444 are securely placed in the vehicle *Vusr*. Therefore, positions *A1* to *A4* of the active sensors 441 to 444 from which the ultrasonic waves, for example, are emitted can be all defined by coordinates (*xa1*, *ya1*) to (*xa4*, *ya4*) that are known in the 2D coordinate system. Also, angles $\phi 1$ to $\phi 4$ at which the active sensors 441 to 444 emit the ultrasonic waves are known. In ~~this~~ the third embodiment,

the angles $\phi 1$ to $\phi 4$ are formed by the X-axis and the emitted waves, and FIG. 17 exemplarily shows only the angle $\phi 1$. As such, the above coordinates ($x1$, $y1$) is equal to ($d1 \cdot \cos\phi 1 + xa1$, $d1 \cdot \sin\phi 1 + ya1$), and those coordinates ($x2$, $y2$) to ($x4$ to $y4$) are equal to ($d2 \cdot \cos\phi 2 + xa2$, $d2 \cdot \sin\phi 2 + ya2$) to ($d4 \cdot \cos\phi 4 + xa4$, $d4 \cdot \sin\phi 4 + ya4$), respectively.

**[0049]** Then, based on the ~~thus~~ calculated four points *P1* to *P4*, the processor 41 calculates coordinates (*xlmt*, *ylmt*) of the corner point *Pcnr* of the obstacle *Vbst* as one example of the no-go point *Plmt* (step S44). By referring to FIG. 18, the process in ~~FIG. 44~~ step S44 is now described in detail. The processor 41 first performs a Hough transform with respect to the points *P1* to *P4* so that curves *C1* to *C4* are derived in the Hough space which is defined by the $\rho$-axis and $\theta$-axis. Here, the curves *C1* to *C4* are expressed as the following equations (1) to (4), respectively.

$$\rho = x1 \cdot \cos\theta + y1 \cdot \sin\theta \quad \ldots(1)$$

$$\rho = x2 \cdot \cos\theta + y2 \cdot \sin\theta \quad \ldots(2)$$

$$\rho = x3 \cdot \cos\theta + y3 \cdot \sin\theta \quad \ldots(3)$$

$$\rho = x4 \cdot \cos\theta + y4 \cdot \sin\theta \quad \ldots(4)$$

**[0050]** According to the above equations (1) and (2), the processor 41 calculates coordinates ($\rho 1$, $\theta 1$) of an intersection point *Pc1* of the curves *C1* and *C2* in the Hough space, and according to the equations (2) to (4), the processor 41 calculates coordinates ($\rho 2$, $\theta 2$) of an intersection point *Pc2* of the curves *C2* to *C4* in the Hough space. From the intersection point *Pc1*, the processor

41 then derives an equation for a straight line *P1 P2*. Here, the line *P1 P2* is expressed by the following equation (5) on the 2D coordinate system. Similarly, a line *P2 P4* is expressed by ~~an~~ the following equation (6).

$$y = (-\cos\theta 1 \cdot x + \rho 1)/\sin\theta 1 \quad ...(5)$$

$$y = (-\cos\theta 2 \cdot x + \rho 2)/\sin\theta 2 \quad ...(6)$$

From those equations (5) and (6), the processor 41 calculates coordinates of an intersection point of the line *P1 P2* and the line *P2 P3*, and the resulting coordinates are determined as the above-mentioned coordinates (*xlmt*, *ylmt*).

**[0051]** By similarly going through steps S4 and S5 of FIG. 5, the processor 41 receives the current rudder angle $\theta$ of the vehicle *Vusr* (steps S45 and S46).

**[0052]** The processor 41 then calculates, in the 2D coordinate system, coordinates (*xcnt*, *ycnt*) of a center point *Pcnt* (see FIG. 19) of the circle traceable by the vehicle *Vusr* when rotated (step S47). The processor 41 also derives equations for circles *Cr1* and *Cr2*, which are traced respectively by the left- and right-rear wheels *Wr1* and *Wr2*, on each rotation center, of the vehicle *Vusr* when rotated around the center point *Pcnt* (step S48). Here, since the coordinates (*xcnt*, *ycnt*)$_T$ and the equations for the circles *Cr1* and *Cr2* are easily calculated under the well-known Ackermann's model, steps S47 and S48 are not described in detail. Further, the circles *Cr1* and *Cr2* include the left- and right-side trajectories *Pp1* and *Pp2* described above in the first embodiment.

[0053] The processor 41 then derives an equation for a straight line *Llmt*, which passes through the coordinates (*xcnr*, *ycnr*) calculated in step S44, and the coordinates (*xcnt*, *ycnt*) calculated in step S47 (step S49). Herein, the straight line *Llmt* specifies the farthest limit for the vehicle *Vusr* to move without colliding with the obstacle *Vbst*.

[0054] The processor 41 next generates the estimated region *Rpt*, which is a region that is enclosed by the circles *Cr1* and *Cr2* calculated in step S48, the straight line *Llmt* calculated in step S49, and a line segment *Lr12* (step S410). Here, the line segment *Lr12* is the one ~~connecting~~ which connects the rotation centers of the left- and right-rear wheels *Wr1* and *Wr2*.

[0055] By similarly going through steps S1 and S2 of FIG. 5, the processor 41 receives the captured image *Scpt* from the image capture device 4 (steps S411, S412). Based on the captured image *Scpt* and the estimated region *Rpt* generated in step S410, the processor 41 then generates the display image *Sout* on the frame memory. More specifically, the processor 41 deforms the estimated region *Rpt* to the one viewed from the image capture device 4, and renders ~~that~~ the estimated region *Rpf* on the captured image *Scpt*. The resulting display image *Sout* looks ~~as~~ like the one shown in FIG. 14. The processor 41 then transfers the display image *Sout* on the frame memory to the display device 6 ~~for display~~ to be displayed thereon (step S414). Such steps S41 to S414 are repeated until the determination becomes Yes in step S415 to end the

processing of FIG. 16. As such, as the estimated region *Rpt* extends to the no-go point *Plmt*, the driver can instantaneously know the farthest limit to move the vehicle *Vusr*.

**[0056]** In the ~~above~~ first to third embodiments as described above, the image capture device 4 is embedded in the rear-end of the vehicle *Vusr*. ~~This is not restrictive~~ The present invention, however, is not restricted thereto, and the image capture device 4 can be embedded in the front-end of the vehicle *Vusr* ~~will also do~~. Further, the number of image capture devices 4 is not limited to one, and may be more than one depending on the design requirements of the drive assistant devices *Uast1* to *Uast3*.

**[0057]** Still further, in the ~~above~~ above-described embodiments, the captured image *Scpt* is the one on which the left- and right-side trajectories *Pp1* and *Pp2*, the estimated path *Pp*, and the estimated region *Rpt* are rendered. Here, the captured image *Scpt* may be subjected to some image processing by the processors 1, 21, and 41 before having those rendered thereon. Such image processing is typified by processing of generating an image of an area around the vehicle *Vusr* viewed from a virtual viewpoint set high up in the vehicle *Vusr*.

**[0058]** Still further, in the ~~above~~ first to third embodiments described above, the captured image *Scpt* is stored in the frame memory in response to the image capture instruction *Icpt* transmitted from the processors 1, 21, and 41 to the image capture device 4. ~~This is not restrictive~~ The present invention, however,

is not restricted thereto, and the captured image *Scpt* is voluntarily generated by the image capture device ~~4,~~ 4 and then stored in the frame memory. Similarly, the rudder angle $\theta$ may be detected voluntarily by the rudder angle sensor 5 without responding to the detection instruction *Idct* ~~coming~~ originating from the processors 1, 21, and 41.

**[0059]** Still further, in the ~~above~~ third embodiment described above, four active sensors 441 to 444 are placed in the vehicle *Vusr*. The ~~number thereof is not restrictive~~ present invention, however, is not restricted thereto, and ~~may be~~ one or more active sensors may be placed in the vehicle *Vusr*. Here, if only one active sensor is placed in the vehicle *Vusr*, the direction of the lens thereof needs to be dynamically changed so that the angle $\phi$ of the emitted waves is set to be wider.

**[0060]** Still further, in the ~~above~~ above-described third embodiment, the active sensors 441 to 444 are provided herein as one example of a measuring sensor ~~in Claims~~ for measuring the distances *d1* to *d4* to the obstacle *Vbst*. ~~This is not restrictive~~ The present invention, however, is not restricted thereto, and ~~other~~ another type of measuring sensor such as a passive sensor may be used. Here, to structure such an exemplary passive sensor, two image capture devices are required to cover the area to the rear of the vehicle *Vusr*. These image capture devices each pick up an image of the obstacle *Vbst* located behind the vehicle *Vusr*. Based on a parallax of the obstacle in images, the processor 41

then measures a distance to the obstacle *Vbst* with stereoscopic views (stereoscopic vision).

**[0061]** Still further, in the ~~above-~~ above-described embodiments, the programs *PGa* to *PGc* are stored in the rendering devices *Urnd1* to *Urnd3*, respectively. ~~This is not restrictive~~ The present invention, however, is not restricted thereto, and those programs *PGa* to *PGc* may be distributed in a recording medium typified by a CD-ROM, or over a communications network such as the Internet.

**[0062]** While the present invention has been described in detail, the foregoing description is in all aspects illustrative and not restrictive. It is to be understood that numerous other modifications and variations can be devised without departing from the scope of the invention.

ABSTRACT OF THE DISCLOSURE

In a rendering device ~~Urnd1~~, a processor ~~1~~ derives an estimated path to be traced by ~~a~~ the left- and right-rear wheels of a vehicle based on a rudder angle ~~θ~~ that is provided by a rudder angle sensor ~~5~~. The processor ~~1~~ then determines positions for overlaying indicators on the derived estimated path. The processor ~~1~~ then renders the indicators on ~~thus~~ the determined points in a captured image which is provided by an image capture device, and generates a display image. ~~Here, in~~ In the display image, the indicators move along the estimated path in the ~~heading~~ direction ~~of~~ the vehicle is heading towards. In this manner, the estimated path in the display image that is generated by the rendering device ~~Urnd1~~ becomes noticeable for a driver of the vehicle.